Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

CRT-2006-30

A BRANCH-AND-CUT ALGORITHM FOR THE UNDIRECTED PRIZE COLLECTING TRAVELING SALESMAN PROBLEM

by

Jean-François Bérubé
Michel Gendreau
Jean-Yves Potvin

December 2006

Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7 Canada
Téléphone: (514) 343-7575 / Télécopieur: (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

HEC ÉCOLE POLYTECHNIQUE MONTRÉAL Université de Montréal

2007-00278

PUBLICATION
CRT-2006-30

A BRANCH-AND-CUT ALGORITHM FOR THE UNDIRECTED PRIZE COLLECTING TRAVELING SALESMAN PROBLEM

by

Jean-François Bérubé[1]
Michel Gendreau[1]
Jean-Yves Potvin[1]

The research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Fonds québécois de la recherche sur la nature et les technologies (FQRNT).

1. Département d'informatique et de recherche opérationnelle and Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 (berubej, michelg, potvin@umontreal.ca)

Département d'informatique et de recherche opérationnelle - Publication #1292

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2006

Abstract

Given an undirected graph with edge costs and node prizes, the aim of the Prize Collecting Traveling Salesman Problem (PCTSP) is to find a simple cycle minimizing the total edge cost while collecting at least a minimum amount of prizes. In this paper, we present a branch-and-cut algorithm to solve the PCTSP. We have adapted valid inequalities from cuts designed for the Orienteering Problem and we have implemented some existing polyhedral results for the PCTSP. Computational results on instances with more than 500 nodes are reported.

Keywords. Prize Collecting Traveling Salesman Problem, Branch-and-Cut, Valid Inequalities.

Résumé

Le problème du voyageur de commerce avec gains (PVCAG) est défini sur un graphe complet non dirigé où des coûts sont associés aux arêtes et où des gains sont associés aux sommets. Le PVCAG consiste à déterminer un cycle élémentaire de coût minimum permettant d'amasser une quantité de gains prédéterminée. Cet article présente un algorithme de séparation et coupes destiné à résoudre le PVCAG. Nous avons adapté des inégalités valides développées pour le problème d'« orienteering » et nous avons implanté certains résultats théoriques issus d'études polyhédrales du PVCAG. Des résultats numériques obtenus pour des instances de taille allant jusqu'à plus de 500 sommets sont présentés.

Mots clés. Problème du voyageur de commerce avec gains, séparation et coupes, inégalités valides.

Introduction

The Prize Collecting Traveling Salesman Problem (PCTSP) was introduced as a model for scheduling the daily operations of a steel rolling mill [4]. Given an undirected graph with edge costs and node prizes, the aim of the PCTSP is to find a simple cycle minimizing the total edge cost while collecting at least a minimum total prize. Also known as the Quota TSP [2], the PCTSP has been classified as a Traveling Salesman Problem with Profits [11], along with the the Profitable Tour Problem (PTP) and the Orienteering Problem (OP). The PTP, introduced in [10], aims at minimizing the difference between the collected prizes and the travel costs; it is also known as the Simple Cycle Problem [13]. In the OP, one must find a tour that maximizes the total collected prize while maintaining the costs under a fixed value. It has been introduced in a study on orienteering competitions [27] and is also known as the Selective TSP [21] and as the Maximum Collection Problem [20].

Although there are several polyhedral results for the PCTSP [4, 5, 6], they have not been implemented in practice yet. To our knowledge the only exact approach to solve the PCTSP is a branch-and-bound algorithm due to Fischetti and Toth [14]. However, there are many publications on exact methods to solve the OP [11]. In particular, the two branch-and-cut algorithms devised for the symmetric OP in [12, 16] are currently the most effective procedures.

In this paper, we present a branch-and-cut algorithm based on valid PCTSP inequalities derived from the associated knapsack polytope, from the subtour elimination constraints, and from the associated Traveling Salesman (TS) polytope. In Section 1, the mathematical model is presented, along with some notation. Section 2 describes the valid inequalities while separation procedures are discussed in Section 3. The overall branch-and-cut algorithm is described in Section 4. Finally, Section 5 reports computational results with different application orders for the separation procedures, using instances generated from the Traveling Salesman Library TSPLIB [26]. Note that throughout this text, we assume that the reader is familiar with the branch-and-cut methodology, as described in [19].

1 Mathematical model

Let $G = (V, E)$ be an undirected complete graph, with edge set E and vertex set V, among which vertex 1 stands for the depot. The nonnegative integer prizes are denoted p_v for each $v \in V$ ($p_1 = 0$) and the minimum collected prize is denoted $\bar{p}$. For each $e \in E$, the nonnegative integer travel costs c_e satisfy the triangle inequality. We define $E(S) = \{(u, v) \in E : u \in S, v \in S\}$ and $\delta(S) = \{(u, v) \in E : u \in S, v \notin S\}$ for $S \subset V$, and $V(T) = \{v \in V : T \cap \delta(\{v\}) \neq \emptyset\}$ for $T \subseteq E$. We also define, $V' = V \setminus \{1\}$ and for each $v \in V$, we write $\delta(v)$ instead of $\delta(\{v\})$. The decision variables are :

$$x_e = \begin{cases} 1 & \text{if edge } e \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

$$y_v = \begin{cases} 1 & \text{if vertex } v \text{ is visited} \\ 0 & \text{otherwise} \end{cases}$$

Finally, we define $y(S) = \sum_{v \in S} y_v$ and $p(S) = \sum_{v \in S} p_v$ for $S \subseteq V$, and $x(T) = \sum_{e \in T} x_e$ and $c(T) = \sum_{e \in T} c_e$ for $T \subseteq E$. The PCTSP can then be formulated as the following 0-1 integer linear program (LP):

$$\min \sum_{e \in E} c_e x_e \tag{1}$$

subject to

$$\sum_{v \in V} y_v p_v \geq \bar{p} \tag{2}$$

$$x(\delta(v)) = 2y_v \qquad (\forall\, v \in V) \tag{3}$$

$$x(\delta(S)) \geq 2y_v \qquad (\forall\, S \subset V:\ 1 \in S,\ v \in V \setminus S) \tag{4}$$

$$y_1 = 1 \tag{5}$$

$$x_e \in \{0,1\} \qquad (\forall\, e \in E) \tag{6}$$

$$y_v \in \{0,1\} \qquad (\forall\, v \in V') \tag{7}$$

Constraint (2) imposes a minimum bound on the collected prize. It can also be formulated as

$$\sum_{v \in V} p_v (1 - y_v) \leq U \tag{8}$$

where $U = p(V) - \bar{p}$. The degree constraints (3) insure that a feasible solution goes exactly once through each visited vertex. The Subtour Elimination Constraints (SEC) (4) require for each visited vertex $v \in V'$ of a feasible solution to be reachable from the depot by two edge-disjoint paths. In addition to the standard formulation (4), SEC can be written as:

$$x(E(S)) \leq y(S) - y_v \quad (\forall\, S \subset V:\ 1 \in S,\ v \in V \setminus S) \tag{9}$$

or

$$x(E(S)) \leq y(S) - y_v \quad (\forall\, S \subset V:\ 1 \in V \setminus S,\ v \in S) \tag{10}$$

as long as the degree constraints (3) hold. Constraint (5) forces the depot to be visited and constraints (6) and (7) impose that all variables be 0-1. Note that the model forces all feasible solutions to visit at least 3 vertices. Since solutions with less than 3 vertices can be easily found by explicit enumeration, we can assume, without loss of generality, that optimal solutions contain at least 3 vertices.

2 Valid inequalities for the PCTSP

We use inequalities obtained from the associated knapsack polytope, from a combination of the SECs and the minimum prize constraint and from the associated TS polytope. Most of them are derived from the polyhedral results of Balas for the asymmetric PCTSP [4, 5, 6] or are adaptations of valid OP inequalities found in branch-and-cut procedures [12, 16]. To our knowledge, none of these inequalities has ever been implemented to solve the PCTSP.

2.1 Inequalities from the associated knapsack polytope

We consider two types of inequalities based on knapsack constraints. The first is based on the minimum prize constraint (8) and is directly related to the associated knapsack polytope, while the second uses an upper bound on the optimal cost value to define a cost-based knapsack constraint. We will refer to these inequalities as the *cover* and *cost-cover* inequalities, respectively.

2.1.1 Cover inequalities

From the minimum prize constraint (8), we define the associated knapsack polytope $KP = conv\{y \in \{0,1\}^V : y \text{ satisfies (8)}\}$, where $conv$ stands for the convex hull. Among the known classes of valid inequalities for the knapsack problem, we consider the cover inequalities :

$$\sum_{v \in S}(1 - y_v) \leq |S| - 1 \tag{11}$$

where S is a minimal cover for (8), i.e., S is a minimal subset of V such that $p(S) > U$. Equation (11) can be strengthened as :

$$\sum_{v \in (S \cup S')} (1 - y_v) \leq |S| - 1 \tag{12}$$

where $S' = \{v \in V \setminus S : p_v \geq \max_{w \in S} p_w\}$ [3]. The coefficients of the y variables can be lifted to obtain *lifted-cover* inequalities that reinforce equation (12).

Theorem 1 ([3, 7]). *Let S be a minimal cover for (8), $S' = \{v \in V \setminus S : p_v \geq \max_{w \in S} p_w\}$, and S_h the set of the first h elements of S ($\forall i \in S$, $p_i \geq p_{i+1}$ is assumed), $h = 1, ..., |S|$. Let V be partitioned into $V_0, V_1, .., V_q$, $q = |S| - 1$, where:*

$$\begin{aligned} V_h = \left\{v \in (S \cup S') : p(S_h) \leq p_v < p(S_{h+1})\right\}, \quad h = 2, ..., q \\ V_1 = (S \cup S') \setminus \cup_{h=2}^{q} V_h, \\ V_0 = V \setminus (S \cup S') \end{aligned} \tag{13}$$

and define:

$$\pi_v = h, \quad \forall v \in V_h, \quad h = 0, ..., q. \tag{14}$$

Then the inequality:

$$\sum_{v \in S}(1 - y_v) + \sum_{v \in V \setminus S} \pi_v (1 - y_v) \leq |S| - 1 \tag{15}$$

is satisfied by all $y \in KP$.

Since the PCTS polytope is included in KP [4], the lifted-cover inequalities are also valid for the PCTSP.

2.1.2 Cost-cover inequalities

Let c_U be the upper bound on an optimal solution. Then $\sum_{e \in E} c_e x_e \leq c_U$ defines a knapsack constraint in terms of the costs that can be used to derive valid inequalities. Let $S \subseteq V$, $1 \in S$ and σ_S a lower bound on the optimal TSP value on S. Then, if $\sigma_S > c_U$ and if the costs satisfy the triangle inequality,

$$y(S) \leq |S| - 1 \tag{16}$$

is valid for the PCTSP. We are interested in special cases of equation (16) that are easier to separate because they do not require solving a TSP to find the lower bound σ_S. When $|S| = 3$:

$$y_u + y_v \leq 1 \quad \forall u, v \in V' \quad \text{such that} \quad c_{(1,u)} + c_{(u,v)} + c_{(v,1)} > c_U \tag{17}$$

and when $|S| = 2$:

$$y_v = 0 \quad \forall v \in V' \quad \text{such that} \quad 2c_{(1,v)} > c_U \tag{18}$$

2.2 Inequalities from the SEC and knapsack constraint

We now introduce the *cycle-cover* and *conditional* inequalities that both use a knapsack constraint (either on the collected prizes or on the costs) to formulate strengthened versions of the SECs.

2.2.1 Cycle-cover inequalities

The cycle-cover inequalities exploit the minimum prize constraint and the fact that a feasible solution must be a cycle. Let $S \subset V$, $1 \in S$ such that $p(S) < \bar{p}$, then

$$x(E(S)) \leq y(S) - 1 \tag{19}$$

is a valid inequality for the PCTSP. Since $1 \in S$, at least one vertex of S (the depot) must be visited by any feasible solution. Also, since $p(S) < \bar{p}$, a feasible solution cannot be completely contained in S. Therefore, the optimal solution will visit a subset of $1 < n^* < |S|$ vertices of S. Considering the optimal solution, we have $y^*(S) = n^*$. Moreover, since the optimal solution is not completely contained in S, $x(E(S)) \leq n^* - 1$. Hence $x(E(S)) \leq y(S) - 1$.

2.2.2 Conditional inequalities

An upper bound c_U on the objective value can be used to derive inequalities similar to the cycle-cover, but based on the selected edges. Although they are not guaranteed to be valid, these inequalities can be conditionally used in a cutting-plane context. Let $T \subseteq E$ such that $c(T) > c_U$, then

$$x(T) \leq y(V(T)) - 1 \tag{20}$$

is valid for the PCTSP if no feasible solution of value lower than c_U is contained in T, since $x(T) \leq y(V(T))$ holds for every feasible solution. This occurs, in particular, when T defines a simple cycle that goes through the depot and for which $c(T) > c_U$.

2.3 Comb inequalities

Many valid inequalities for the TSP can be adapted to the PCTSP. Among those, we consider comb inequalities [17, 18] that have been adapted to the PCTSP in [5, 6]. Let us consider two sets of vertices, the handle $H \subset V$ and the teeth $T_j \subset V$ $(j = 1, ..., t)$. The general comb inequalities are formulated as:

$$x(E(H)) + \sum_{j=1}^{t} x(E(T_j)) \leq y(H) + \sum_{j=1}^{t} |T_j| - \frac{3t+1}{2} \tag{21}$$

for all $H, T_1, ..., T_t$ satisfying:

$$|T_j \cap H| \geq 1 \qquad j = 1, ..., t \tag{22}$$

$$|T_j \setminus H| \geq 1 \qquad j = 1, ..., t \tag{23}$$

$$T_i \cap T_j = \emptyset \qquad 1 \leq i < j \leq t \tag{24}$$

$$t \geq 3 \text{ and odd} \tag{25}$$

In the special case where

$$|T_j \cap H| = 1 \tag{26}$$

for all j, the inequalities are referred to as *simple comb* inequalities (or Chvátal inequalities) [8]. Simple comb inequalities become *2-matching* inequalities if

$$|T_j \setminus H| = 1 \tag{27}$$

for all j.

2.4 Logical inequalities

Obviously, if an edge $e \in \delta(v)$ is part of a solution, the vertex v must be visited, hence the following logical inequality:

$$x_e \leq y_v \quad \forall e \in \delta(v), \quad v \in V' \tag{28}$$

Note that logical inequalities are a special case of SECs (4) when $e = (j, v) \notin \delta(1)$.

2.5 Summary

Table 1 summarizes the valid inequalities (other then SECs) used in our branch-and-cut algorithm. As mentioned earlier, most of them come from previous polyhedral results on the asymmetric PCTSP or have been adapted from valid OP inequalities.

- *LC*. The cover inequalities are introduced in [3, 7] for the knapsack problem and are used in [12] to solve the symmetric OP. The formulation is analog to (12), but is in terms of edge variables (x) because the knapsack constraint is defined on the total travel cost in the OP. The lifted-cover inequalities are also presented in [3, 7], but have never been implemented to solve neither the PCTSP nor the OP.

Category	Description	References	Equations
Knapsack polytope	Lifted-cover (LC)	[3, 7]	(15)
	Cost-cover (CC)	[16, 22]	(17), (18)
SEC and knapsack	Cycle-cover (CY)	[12]	(19)
	Conditional (CO)	[12]	(20)
TS polytope	2-Matching (2M)	[5, 6, 12, 16]	(21) with (26), (27)
	Simple comb (SC)	[5, 6]	(21) with (26)
Other	Logical (LO)	[12, 16, 22]	(28)

Table 1: Summary of valid inequalities

- *CC.* Cost-cover inequalities are used to solve the symmetric OP in [16, 22] with an upper bound based on the maximum cost constraint.
- *CY.* A formulation similar to our cycle-cover inequalities is used to solve the OP in [12]. The knapsack constraint is defined on the costs and S is defined as a set of edges that do not satisfy the maximum cost constraint.
- *CO.* Conditional inequalities are introduced in [12], again to solve the OP. A lower bound on the total collected prize is used to find infeasible sets of vertices.
- *2M, SC.* Adaptations of the comb inequalities to the PCTSP can be found in [5, 6]. Only the 2-matching inequalities have been implemented in branch-and-cut algorithms for the symmetric OP [12, 16].
- *LO.* Logical inequalities are introduced in [22] and have been used to solve the symmetric OP in [12, 16, 22].

3 Separation algorithms

In this section, we describe the separation procedures used in our branch-and-cut algorithm to identify violated inequalities. In the following, we denote $G^* = (V^*, E^*)$ the graph associated with the solution (x^*, y^*) obtained through the linear relaxation LP, where $V^* = \{v \in V : y_v^* > 0\}$ and $E^* = \{e \in E : x_e^* > 0\}$.

3.1 Subtour elimination constraints

Let us consider x_e^* as the capacity value of edge $e \in E^*$. For any $v \in V^* \setminus \{1\}$, a minimum-capacity$(1, v)$-cut is found in G^* through a maximum-flow algorithm, where the nodes $v \in V^* \setminus \{1\}$ are considered in

decreasing order of the corresponding y_v^*. We used the highest label preflow-push algorithm [1] which runs in $O(|V^*|^2\sqrt{|E^*|})$ for any given node. Hence, we obtain an overall complexity of $O(|V^*|^3\sqrt{|E^*|})$ to find a most violated SEC for each $v \in V^* \setminus \{1\}$. Let $(S_v, V^* \setminus S_v)$ be the minimum-capacity cut obtained for v, with $1 \in (V^* \setminus S_v)$. The set S_v gives a potentially violated SEC (10). When a violated SEC is found, x_v^* is increased by $2 - \sum_{e \in \delta(S_v)} x_e^*$ to prevent the same cut from being generated in subsequent iterations. The set $(V^* \setminus S_v)$ is used to produce potentially violated cycle-cover inequalities (19), as explained later. This separation algorithm is very similar to the one used in [12], although ours has a lower complexity on sparse graphs like G^*, due to the maximum-flow algorithm that we use.

3.2 Lifted-cover inequalities

Lifted-cover inequalities (15) result from the lifting procedure described in Theorem 1. A most-violated cover inequality is first produced from a minimum cover set $S \subset V$ such that $p(S) > U$, and its coefficients are lifted afterward. To obtain a violated cover inequality, the set S should minimize $\sum_{v \in S} y_v^*$, hence the following 0-1 knapsack problem:

$$\alpha^* = \min \sum_{v \in V} y_v^* z_v \tag{29}$$

subject to:

$$\sum_{v \in V} p_v z_v \geq U \tag{30}$$

$$z_v \in \{0, 1\} \qquad (\forall\, v \in V) \tag{31}$$

Obviously, variables z_v for which the corresponding y_v^* is equal to 0 can be fixed to 1 and those for which the corresponding y_v^* is equal to 1 can be fixed to 0. If $\alpha^* < 1$, S gives a most-violated cover inequality. However, the latter can be strengthened by making S a minimal cover set, which can be done with a simple greedy algorithm. We efficiently solve the 0-1 knapsack problem with our implementation of Martello and Toth specialized branch-and-bound algorithm [23, 24].

3.3 Cost-cover inequalities

Violated cost-cover inequalities (17) are identified through simple enumeration with complexity $O(|E|)$. Inequalities (18) are all added to the LP during its initialization at the root node of the branching tree through another enumeration procedure that runs in $O(|V|)$.

3.4 Cycle-cover inequalities

The separation algorithm for the cycle-cover inequalities (19) is embedded within the separation algorithm of the SECs. As already mentioned, we use the minimum-capacity cut produced by the maximum-flow algorithm to obtain an hopefully violated cycle-cover inequality. Therefore, one simply needs to check for violation before adding a cycle-cover inequality.

3.5 Conditional inequalities

The heuristic separation algorithm for the conditional inequalities (20) enumerates the simple cycles of G^* that start from the depot. Whenever the cost of a cycle is greater then the best known upper bound on the optimal solution, the cycle is considered as a potential candidate for T and the inequality is checked for violation. Obviously, enumerating all simple cycles of G^* might be very long. We thus limit the number of enumerated cycles to 100.

3.6 Comb inequalities

We consider both 2-matching and simple comb inequalities. The 2-matching inequalities are separated through the heuristic procedure suggested in [12] while violated simple comb inequalities are identified through the heuristic procedure 5.3 of Padberg and Rinaldi [25]. Although the 2-matching inequalities are a special case of the simple comb inequalities, it is worth having two distinct separation procedures due to their heuristic nature. With two different heuristics, it is hoped that different 2-matching inequalities will come up.

3.7 Logical inequalities

Logical inequalities are found by complete enumeration with a complexity of $O(|E^*|)$.

4 Overall branch-and-cut algorithm

Figure 1 depicts the overall branch-and-cut algorithm, where the main phases are represented with boxes. Explanations for each phase follow, along with details on some parameters.

4.1 Initial solution

At the root node of the branching tree, a feasible solution is computed. The cost of this solution gives an upper bound on the optimal solution value (UB) and the solution itself will be used to remove some variables from the initial LP as explained later. The initial solution is found through a heuristic algorithm based on the *GENIUS* [15] and *Extension and Collapse* [9] procedures. Developed for the TSP, *GENIUS* is made of two phases: a tour construction phase based on generalized insertions (*GENI*) and a post-optimization phase (*US*). The *Extension and Collapse* heuristic is an improvement procedure for the PCTSP based on an iterative execution of two consecutive phases: the *Extension* phase and the *Collapse* phase. In the *Extension* phase, the collected prize of a feasible solution is improved by adding nodes to the current cycle. The cost of the resulting extended solution is then reduced in the *Collapse* phase, where the most expensive nodes might be removed.

Our heuristic algorithm starts with a set $V_S = \{v \in V : p(V_S) \geq p\}$ found through a greedy procedure that selects the nodes in random order until the minimum prize constraint is satisfied. The following steps are then applied:

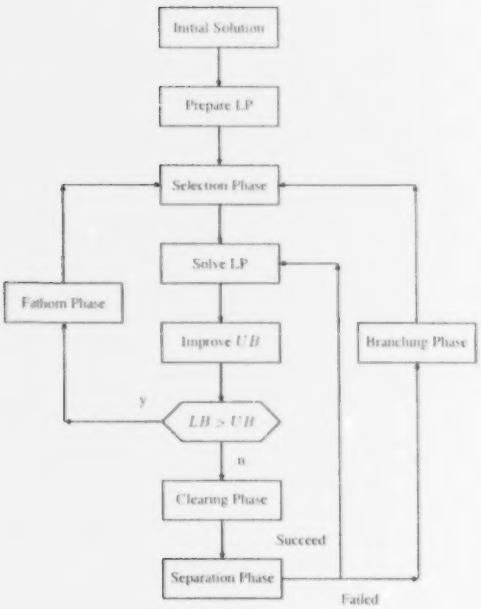

Figure 1: Overall branch-and-cut algorithm: the algorithm begins with the construction of the initial solutions and ends when the selection phase failed, that is when there is no more branching node to explore.

1. A feasible solution S_k visiting all nodes in V_S is constructed and improved with *GENIUS*.
2. Starting with S_k, the following steps are applied until no more improvement is possible:
   (a) Apply the *Extension phase* as described in [9] to obtain solution S_e.
   (b) Apply *GENIUS* on the nodes of S_e to obtain S'_e and keep the best of S_e and S'_e.
   (c) Apply the *Collapse phase* as described in [9] to obtain S_c.
   (d) Set S_k as the best of S_k and S_c.
3. Apply *GENIUS* on the nodes of the best known solution S_k to obtain S'_k and keep the best of S_k and S'_k.

4.2 Initial linear program

The initial LP (*Prepare LP* box of figure 1) includes equations (1), (2), (3) and (5) along with all the node variables (y), but only a fraction of the edge variables (x). This reduces the size of the LP and thus helps

to solve it efficiently. Edge variables visited by the initial solution are included in the LP along with the 5 lowest cost edges incident to each node visited in the initial solution.

Because some edge variables are removed from the initial LP, we cannot fathom branching nodes for which the LP is infeasible because it might be feasible if some previously excluded variables were present. In order to prevent the LP from being infeasible, we add a virtual vertex ω and artificial edges from ω to each node $v \in V$. In the objective of the LP, the cost associated with artificial edge a is given by: $c_a = c(E) + 1$. The artificial variables (z_a) must also be included in the degree constraints (3) and may take integer values 0, 1 or 2. Suppose, for example, that a node v is visited ($y_v > 0$) while all variables that correspond to its incident edges are not part of the LP. Let us assume that $0.5 < y_v \leq 1$. Then, the corresponding artificial edge variable (v, ω) needs to be: $1 < z_a \leq 2$ to satisfy the degree constraint.

4.3 Selection phase

The iterative part of the algorithm starts with the selection phase where the next branching node to be explored in selected. A priority queue maintains the nodes in non-decreasing order of their associated LP solution value (LB). The selection consists in removing a node with the lowest LB from the queue. Usually, the latter node is selected but, in some cases, the y variables selected for branching are such that the maximum collected prize is lower then $\bar{p}$ and so the node is automatically fathomed. When the queue is empty, the algorithm stops and the best feasible solution is guaranteed to be the optimal solution.

4.4 Solving phase

The LP is solved with the CPLEX commercial software. However, it is important to observe that the LP solution value may not be a valid lower bound because some variables have been excluded from the model (see *Initial LP* and *Clearing Phase*). A pricing operation is performed to check if some excluded variables should be integrated into the LP: excluded variables with negative reduced costs are added to the LP before it is re-optimized. When all excluded variables have a positive reduced cost, the solution value of the LP is guaranteed to be a valid lower bound. Note that no more than 100 variables are added to the LP in a single pricing operation to keep the LP as small as possible.

4.5 Improving the upper bound (UB)

Because the UB allows us to prune the branching tree, its improvement might speed up the algorithm by reducing the number of explored nodes. To improve the quality of this bound, the following heuristic procedure is applied after 5 consecutive executions of the separation phase.

1. Node selection: nodes are sorted in decreasing order of their corresponding y^* value and are selected in turn, until $\bar{p}$ is reached.

2. Solution construction: a solution is then constructed with *GENIUS* [15].

If this new feasible solution is better then the best known feasible solution, the UB is updated.

4.6 Clearing phase

The aim of the clearing phase is to keep the size of the LP as small as possible by reducing the number of edge variables (x). To do so, we remove all variables x_e with $\lfloor LB + 0.2r_e \rfloor \geq UB$, where LB is the current LP value and r_e is the reduced cost of x_e. Another way to reduce the size of the LP is to remove inactive constraints. We do so for constraints that have not been tight for the last 30 iterations. All removed constraints are stocked in a constraint pool.

4.7 Separation phase

This is where violated inequalities are added to the LP. If no violated cut is found in the constraint pool, the separation algorithms described in the previous section are applied sequentially until one succeeds. The violated cuts are then added and the separation phase ends. The order in which the separation algorithms are executed varies. We have tried several sequences which are described in the section on computational results. However, we should mention that all separation sequences are not possible because some separation algorithms assume that all violated SECs have been added to the LP. This is the case of the lifted-cover, cycle-cover, 2-matching, simple comb and conditional inequalities.

Adding violated cuts does not always improve the LP solution value. This phenomenon often called *tailing off* can be reduced by forcing the algorithm to branch even if some more violated cuts might be found. We force the execution of the branching phase whenever the LP solution value does not improve by at least 0.01 in the last 5 iterations on the same branching node.

4.8 Branching phase

The algorithm always branch on the most fractional y variable, that is the y^* which is closer to 0.5. If all the y variables are integer, then we branch on the most fractional x variable.

Because some cuts might be removed in the clearing phase, it is possible that the LP-solver finds a solution with a lower value than the lower bound associated with the branching node. To prevent this, each branching node keeps trace of the active cuts that have produced its lower bound. These cuts are added to the LP-model (if they are not already there) when a node is selected in the selection phase.

4.9 Fathom phase

The fathom phase corresponds to fathoming the current branching node.

5 Computational results

Polyhedral theory and existing branch-and-cut algorithms for TSP, OP and knapsack problem give a good insight on the kind of inequalities that should be considered and on the useful features of a good branch-and-cut implementation for the PCTSP. However, an analysis of the behavior of the various inequality families

would help future cutting-plane algorithms for the PCTSP. To this aim, we give results for many versions of the separation phase, as explained later. The branch-and-cut algorithm was implemented in C++ and was run on AMD Opteron 2.2 Ghz processors. The LPs were solved using CPLEX 9.3 and a time limit of 4 hours (14400 seconds) was imposed on each instance.

5.1 Instances

We transformed VRP and TSP instances of the TSPLIB [26] into PCTSP instances using the rules provided in [12]. We considered all instances with 70 to 532 nodes for which the node coordinates were available. For VRP instances, the demands are interpreted as the node prizes. For TSP instances, the prizes p_v ($v \in V'$) are generated in three different ways:

- Generation 1: $p_v = 1$;
- Generation 2: $p_v = 1 + (7141v + 73) \mod 100$;
- Generation 3: $p_v = 1 + \lfloor 99\frac{c_{1,v}}{\theta} \rfloor$, where $\theta = \max_{w \in V'} c_{1,w}$.

Instances of generation 1 are in general easy problems since all prizes are the same. Generation 2 produces instances with pseudo-random prizes while generation 3 produces hard problems where larger prizes are associated with vertices that are further from the depot. For each instance, we search for the optimal solution of problems with a minimum collected prize $\bar{p} = \alpha p(V)$, with $\alpha \in \{0.25, 0.50, 0.75\}$.

5.2 Separation sequences

Many versions of the separation phase have been tried out to determine the best sequence. A separation sequence is defined by the order in which the inequality families are separated. In the next sections, we present results for the five most interesting separation sequences corresponding to letters A, B, C, D and E, where:

A: LO, CY[1];

B: LO, CY, CC, LC;

C: LO, CY, 2M, SC;

D: LO, CY, CC, LC, 2M SC;

E: LO, CY, CC, LC, 2M, SC, CO.

[1] In the following, CY includes both SECs and cycle cover inequalities because they are separated through the same algorithm.

5.3 Computation time for each sequence

For each instance and for each value of α, we show the results for each separation sequence. Tables 2, 3, 4 and 5 show the computation time in seconds needed to solve the VRP instances and to solve all three generations of TSP instances, respectively. For each instance, the fastest separation sequence is identified in bold face and the letters *t.l.* (time limit) mean that the instance cannot be solved within the time limit.

In the following, we use two statistics to determine the quality of a separation sequence over a group of instances. They are noted as the pair (η_i, γ_i) where η_i is the ratio of the number of times sequence i is the fastest, on the size of the group and γ_i represents the average gap over each instance of a group, between the computation time of sequence i and the fastest time.

For VRP instances, the best separation sequence is B (40%, 15%) while C (0%, 45%) is the worst. This observation suggests that comb inequalities are ill suited to VRP instances while cover inequalities seem to be very useful. It is exactly the opposite for TSP generation 1 instances where sequences B (20%, 63%) and C (32%, 15%) are the worst and best sequences, respectively. This is not a surprise since cover inequalities are always satisfied when all prizes are equal, which is the case for generation 1 instances. The results are more ambiguous for generation 2 instances. Sequence A (26%, 205%) is the best according to η_i, but it is the worst in terms of γ_i. In general, we observe large variations in γ_i when comb inequalities are not separated (sequences A and B (22%, 203%)). The best compromise appears to be separation sequence D (18%, 32%). Finally, for generation 3 instances, the cover inequalities reduces dramatically the variations in γ_i. It goes down from 750% for sequence A (18%, 750%) to 21% for sequence D (21%, 21%).

Trends also emerge when we look at the data in terms of the α parameter instead of the instance type. If η_i remains relatively the same for each sequence, γ varies a lot. For all three values of α, the largest variations are observed with sequence A (from 190% to 385%). When $\alpha = 0.25$, adding the cover inequalities reduces γ_i to less then 27%. To obtain a similar stability, one needs to add both cover and comb inequalities when $\alpha = 0.5$. Finally, for $\alpha = 0.75$, the comb inequalities alone suffice to reduce dramatically the variations.

In general, separation sequences A, B and C do not seem appropriate for the PCTSP, because of their highly unpredictable behavior. However, sequence D which combines both cover and comb inequalities is much more robust. Separation sequence E (sequence D plus the conditional inequalities) shows results similar to sequence D, but the latter is often slightly better.

α	0.25					0.50					0.75				
Instance	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
att48	0.67	0.54	0.73	**0.52**	0.54	0.46	**0.45**	0.47	0.47	0.47	0.44	0.45	0.46	**0.44**	0.45
eil51	**0.29**	0.32	0.33	0.32	0.31	**1.3**	1.32	1.44	1.39	1.53	3.2	**3.03**	4.48	4.57	5.06
eilA76	5.54	**4.11**	5.49	4.75	4.85	3.75	3.92	3.85	4.11	**2.97**	11.11	9.03	11.68	**4.88**	10.31
eilA101	9.17	7.39	9.86	7.49	**7.33**	12.43	13.26	13.23	**8.53**	8.64	7.87	**4.99**	7.55	8.35	5.24
gil262	507.75	**425.49**	691.4	444.75	448.97	918.06	**534.61**	841.27	655.34	566.89	319.19	353.03	508.08	**278.45**	322.29

Table 2: Computation times (in seconds) VRP instances

α	0.25					0.50					0.75				
Instance	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
st70	0.63	0.65	0.63	0.64	**0.6**	0.89	0.89	0.92	0.91	**0.85**	**3.14**	3.18	9.59	9.76	5.26
eil76	0.8	0.85	**0.8**	0.93	0.88	1.0	1.01	**0.94**	1.09	1.03	0.78	**0.78**	0.84	0.83	0.82
pr76	1.46	1.39	1.38	1.42	**1.31**	2.7	2.83	2.53	**2.39**	2.46	834.84	867.67	110.32	113.17	**109.56**
gr96	6.24	5.41	6.77	5.58	**5.21**	42.77	**38.05**	54.13	45.46	56.39	**12.5**	12.88	15.31	14.4	14.51
rat99	3.11	2.93	**2.87**	3.12	3.13	18.23	20.03	15.32	**14.3**	15.02	12.84	12.83	**11.67**	11.74	12.06
kroA100	13.11	13.18	13.3	**13.07**	14.13	9.86	10.47	9.26	**9.11**	9.33	15.91	17.01	11.73	**10.83**	16.9
kroB100	10.19	11.01	11.84	11.78	**10.16**	7.61	8.06	6.2	**5.72**	5.94	8.93	9.5	7.79	8.24	**7.42**
kroC100	26.37	19.09	24.87	**16.99**	19.4	**18.26**	19.11	19.27	18.58	21.65	**10.11**	11.3	10.14	11.1	10.37
kroD100	20.21	**16.11**	26.74	19.3	26.67	**6.3**	6.31	6.31	6.36	7.0	7.19	7.19	**6.99**	7.38	7.04
kroE100	8.33	**6.75**	7.77	7.12	6.8	11.82	11.93	**6.87**	6.89	7.41	8.08	8.13	**7.12**	7.13	7.57
rd100	**4.57**	4.6	4.61	5.0	4.69	**6.53**	6.54	9.05	9.15	9.39	**9.52**	9.59	12.98	14.36	14.99
eil101	5.01	4.8	**4.59**	4.98	5.57	**4.36**	4.57	6.91	6.99	6.85	3.4	3.42	**3.1**	3.11	3.35
lin105	**7.79**	8.75	8.79	8.7	9.99	**168.02**	170.79	406.74	462.42	461.7	248.64	289.41	**103.17**	105.7	113.1
pr107	2.41	2.44	**2.34**	2.43	2.43	7.32	6.9	6.9	7.61	**6.87**	17.25	**16.23**	17.05	17.44	18.95
pr124	70.38	35.55	58.65	31.91	**30.38**	**13.3**	14.79	13.35	13.61	13.37	68.33	72.23	44.14	48.72	**42.17**
bier127	6.7	**6.5**	6.81	7.18	6.97	4.79	4.59	**4.49**	4.76	4.59	12.16	**11.1**	13.47	14.95	14.26
ch130	**20.4**	21.82	22.92	31.87	39.01	9.02	8.64	8.67	9.15	**8.64**	76.91	**69.65**	88.51	82.78	84.4
pr136	**20.21**	20.67	20.7	23.35	22.81	124.64	139.49	79.73	**71.88**	72.55	27.55	27.67	26.59	25.12	**24.83**
gr137	17.93	18.91	19.42	**17.58**	20.02	10.67	11.28	**10.62**	11.44	10.64	25.17	25.52	**23.3**	23.42	25.62
pr144	29.2	29.1	**29.09**	29.2	32.53	95.37	**84.26**	101.91	102.69	93.19	**138.9**	142.83	158.99	160.1	167.86
ch150	22.45	**20.31**	20.52	20.63	23.77	23.51	**22.95**	29.18	32.11	31.9	775.83	736.96	**132.18**	141.12	144.97
kroA150	22.71	21.47	**21.35**	22.91	23.19	4129.76	3741.47	**2137.76**	2413.69	2606.57	528.8	539.57	**329.77**	338.66	571.73
kroB150	47.14	48.88	42.39	42.35	**39.98**	43.7	43.3	38.89	43.39	**36.53**	27.62	27.58	26.03	**25.68**	27.4
pr152	51.3	50.52	**44.14**	44.4	51.32	**68.85**	71.55	89.39	90.31	85.21	**161.66**	170.97	220.71	224.48	214.31
u159	163.98	**90.94**	165.77	97.05	96.31	**570.01**	587.9	669.22	681.56	1131.74	**48.3**	48.71	49.39	49.31	50.13
rat195	145.39	163.57	161.19	165.09	**102.23**	167.31	**156.26**	193.9	217.41	212.19	784.06	791.88	219.06	**211.16**	240.82
d198	52.07	55.0	**46.58**	49.14	48.58	2376.91	2477.24	**1366.88**	1639.17	1924.25	5138.72	5122.82	254.54	**222.57**	250.56
kroA200	201.84	**197.99**	201.95	270.01	296.07	122.58	141.39	125.16	**118.79**	153.39	68.36	75.93	**63.84**	69.2	70.7
kroB200	70.38	69.69	91.74	93.06	**64.5**	370.7	**351.33**	438.5	446.32	418.92	66.94	77.72	**61.29**	61.59	61.35
gr202	**30.33**	30.46	30.35	33.16	30.89	**328.51**	356.99	354.96	418.53	433.1	774.43	691.95	**108.26**	110.17	115.38
ts225	8859.2	7367.46	8036.32	10433.8	**6531.68**	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
tsp225	247.85	223.06	**187.05**	213.39	189.28	842.24	917.27	363.39	**317.29**	368.79	t.l.	t.l.	1677.81	**1522.61**	1950.21
pr226	495.54	489.16	445.35	467.35	**410.39**	**5429.52**	5679.24	11592.8	12845.3	t.l.	575.93	632.57	604.23	600.84	**573.84**
gr229	89.34	95.71	92.65	95.53	**83.23**	**154.54**	155.54	180.22	164.81	179.53	180.18	215.32	165.87	**143.54**	162.05
gil262	184.15	200.71	218.12	**179.51**	189.27	169.97	**165.14**	215.19	203.0	228.76	t.l.	t.l.	t.l.	t.l.	t.l.
pr264	24.85	25.74	25.25	27.08	**24.12**	610.36	615.39	606.5	616.03	**532.23**	196.23	193.53	**177.51**	184.86	180.81
lin318	t.l.	t.l.	t.l.	t.l.	t.l.	2945.12	2362.47	2772.84	2411.95	**2355.21**	**1860.79**	1928.18	3288.0	3558.68	3301.03
rd400	525.68	**483.4**	554.11	563.57	520.11	2287.49	2767.43	**2110.73**	2309.27	2157.72	t.l.	t.l.	**11466.7**	11624.2	11862.1
fl417	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
gr431	1295.98	1138.62	954.74	**883.06**	1042.92	t.l.	t.l.	**14285.7**	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
pr439	t.l.	t.l.	t.l.	t.l.	t.l.	1711.1	1707.27	**1483.28**	1600.67	1707.12	t.l.	t.l.	t.l.	t.l.	t.l.
pcb442	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
d493	12747.9	**11933.1**	t.l.	t.l.	t.l.	3184.35	3802.57	**1943.26**	2327.6	1959.04	t.l.	t.l.	t.l.	t.l.	t.l.
att532	t.l.	t.l.	**4521.65**	4568.92	4570.98	t.l.	t.l.	11225.0	12437.2	**10280.1**	t.l.	t.l.	t.l.	t.l.	t.l.

Table 3: Computation times (in seconds) TSP generation 1 instances

α	0.25					0.50					0.75				
Instance	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
st70	0.97	1.02	**0.96**	1.03	1.04	1.73	1.39	1.77	1.43	**1.31**	32.71	39.12	8.68	**7.5**	8.22
eil76	**1.38**	1.47	1.47	1.6	1.49	10.09	8.68	9.99	**6.44**	6.71	5.41	**5.21**	7.77	7.52	8.8
pr76	5.9	6.15	6.14	**5.87**	6.28	**22.13**	24.99	24.91	29.46	27.77	5563.03	5306.9	**188.18**	222.07	237.83
gr96	**34.76**	56.92	51.98	70.24	79.96	37.19	36.87	35.2	**34.6**	35.17	847.3	809.82	**62.01**	65.11	72.38
rat99	**17.75**	24.2	18.02	29.49	27.66	80.73	**61.5**	107.9	85.63	64.84	**24.67**	26.62	30.75	32.09	39.71
kroA100	66.36	45.69	93.53	49.24	**38.38**	**29.7**	38.18	40.39	47.2	48.99	2499.06	1992.84	**96.42**	101.01	107.3
kroB100	29.27	**7.01**	45.52	7.24	8.04	**41.7**	58.1	58.66	64.24	79.21	83.57	98.28	**35.41**	43.93	50.16
kroC100	88.77	80.96	79.05	73.82	**68.63**	**41.16**	41.22	45.66	45.49	43.71	32.63	**29.02**	46.08	36.74	41.46
kroD100	78.87	106.76	**75.45**	101.11	95.51	**30.74**	55.01	36.04	57.27	72.59	**40.37**	49.01	50.38	67.73	61.69
kroE100	**12.23**	14.36	12.84	15.77	12.53	18.88	20.27	**17.9**	20.13	18.11	**91.46**	135.82	114.55	136.36	171.21
rd100	26.2	23.95	**19.87**	25.26	23.69	26.73	37.69	**22.29**	27.53	32.95	44.95	47.55	45.58	**44.44**	48.75
eil101	3.95	3.88	**3.73**	3.82	4.26	**9.11**	17.91	20.54	20.81	19.91	1880.45	2427.11	**71.52**	78.58	73.64
lin105	**5.48**	20.17	6.03	21.08	20.84	1000.36	1391.22	**716.02**	1297.29	1401.34	152.45	**143.45**	281.8	337.33	401.06
pr107	40.1	**22.7**	38.21	23.1	25.39	**76.69**	134.74	111.3	140.59	147.78	219.0	228.36	266.8	**212.35**	279.91
pr124	1005.12	**130.58**	1113.76	139.06	132.04	272.44	**162.39**	230.29	179.91	169.61	402.07	393.02	535.59	**390.93**	396.02
bier127	**50.33**	75.77	54.76	69.57	89.93	**37.55**	50.75	52.19	61.52	70.38	2801.09	3638.71	**298.98**	490.92	395.69
ch130	**79.37**	84.41	81.14	85.43	139.8	89.16	**83.66**	99.76	97.82	144.6	**37.41**	38.29	49.84	48.25	46.15
pr136	**19.61**	26.44	21.71	25.11	70.47	t.l.	t.l.	t.l.	t.l.	t.l.	416.9	**308.91**	481.46	376.08	408.4
gr137	**75.21**	159.8	99.58	194.82	202.78	415.51	385.95	**366.15**	434.28	1121.3	5564.43	5337.74	**441.08**	446.95	993.67
pr144	69.21	**56.94**	61.5	61.45	57.33	325.61	307.93	**284.38**	364.66	386.12	974.1	**963.33**	1522.52	1540.87	1742.65
ch150	**35.57**	46.64	40.26	47.58	49.12	5247.51	4354.05	571.06	557.25	**541.81**	53.51	**36.02**	77.18	103.18	111.41
kroA150	197.09	**172.63**	231.04	208.67	246.71	**60.85**	116.52	83.83	108.19	109.35	t.l.	9616.11	8060.92	**6748.62**	7105.66
kroB150	58.29	81.22	59.73	**56.77**	62.73	497.6	555.35	526.49	**469.95**	597.88	315.9	**238.58**	283.69	269.0	298.28
pr152	134.02	121.83	126.11	122.18	**121.46**	311.2	**249.75**	353.62	279.57	291.23	**4299.57**	5656.18	t.l.	t.l.	t.l.
u159	1305.75	398.07	3370.03	**320.51**	415.25	1328.81	1487.92	1141.27	1261.89	**763.28**	3748.82	**2569.05**	4118.58	2992.23	2625.57
rat195	233.62	184.43	218.73	146.9	**129.58**	176.73	191.36	151.77	**112.81**	164.17	t.l.	t.l.	1019.9	**543.85**	546.54
d198	**141.14**	157.66	167.9	147.99	170.03	t.l.	t.l.	**2579.62**	2687.57	2870.17	t.l.	t.l.	t.l.	t.l.	t.l.
kroA200	1785.19	1852.72	1794.22	1490.67	**1469.1**	t.l.	t.l.	3154.34	2395.47	**2278.69**	t.l.	t.l.	t.l.	t.l.	t.l.
kroB200	228.87	239.42	266.34	**223.31**	325.17	**415.38**	542.56	630.25	678.88	572.06	t.l.	t.l.	t.l.	t.l.	t.l.
gr202	234.49	171.82	223.17	**169.37**	243.85	t.l.	t.l.	853.78	816.21	**753.52**	**636.21**	902.83	890.18	1203.74	1233.42
ts225	**576.52**	695.5	597.62	711.45	912.85	920.47	**907.77**	1100.33	1116.27	1313.77	t.l.	t.l.	2624.75	**2078.69**	2177.38
tsp225	570.56	**341.86**	647.29	381.64	359.68	1845.77	**1803.93**	2580.47	2737.31	3756.25	772.72	**670.7**	995.47	1021.26	1195.95
pr226	606.84	672.3	559.83	663.68	**498.88**	11868.3	9565.39	13787.4	9304.43	**8186.06**	1919.72	3814.77	2372.14	2459.0	**1846.79**
gr229	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	4111.48	**3478.96**	4874.59	**1254.23**	1395.39	1623.29	1774.28	1813.05
gil262	t.l.	t.l.	t.l.	t.l.	t.l.	237.81	182.96	280.99	**165.21**	181.32	t.l.	t.l.	4375.41	4545.85	**4055.05**
pr264	4375.92	3821.61	4216.9	3575.39	**3447.13**	t.l.	t.l.	t.l.	t.l.	t.l.	1997.82	**1311.34**	2120.06	1989.84	1791.26
a280	1589.72	**1413.6**	1543.83	1596.84	1497.26	t.l.	t.l.	t.l.	t.l.	t.l.	588.43	**561.59**	948.92	730.89	840.74
pr299	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
lin318	t.l.	t.l.	t.l.	t.l.	t.l.	3471.54	**3394.98**	4045.62	4216.94	t.l.	5631.19	6226.57	8402.75	**4222.08**	5704.07
rd400	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	**3102.53**	3111.56	3737.95	t.l.	t.l.	10893.0	**6745.16**	7511.47
fl417	6279.9	6364.75	**5488.45**	6368.88	6887.67	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
gr431	t.l.	t.l.	**1882.38**	2025.43	2350.89	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
pr439	11586.6	t.l.	**10242.4**	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
pcb442	6062.16	6348.61	**4992.98**	6079.2	5976.14	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
d493	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
att532	9361.16	12607.7	10537.9	9569.42	**9026.86**	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.

Table 4: Computation times (in seconds) on TSP generation 2 instances

α	0.25					0.50					0.75				
Instance	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
st70	28.66	13.31	25.45	15.9	**13.05**	**7.71**	9.51	9.42	10.54	11.44	222.53	232.93	32.5	**20.62**	22.22
eil76	55.51	**32.68**	52.64	35.84	95.77	9.31	11.36	**9.05**	12.23	16.73	**1.71**	1.82	1.79	1.79	1.91
pr76	2906.35	1148.11	2751.71	**1121.31**	1319.58	25.56	28.95	**24.02**	33.02	25.74	276.54	197.89	**147.24**	158.52	160.3
gr96	170.97	**137.2**	186.51	139.5	162.69	58.8	52.31	67.26	**50.48**	65.73	**272.77**	295.92	349.28	354.68	392.73
rat99	t.l.	1180.39	t.l.	1058.6	**922.85**	64.41	**55.05**	248.0	218.24	289.83	2568.06	2364.28	43.26	39.01	**37.84**
kroA100	361.09	318.52	372.11	296.99	**254.08**	**17.47**	17.97	19.16	18.61	20.28	5380.06	3560.1	**117.14**	160.49	154.27
kroB100	505.63	266.43	549.44	240.78	**200.34**	47.54	67.31	**42.17**	62.85	55.8	1241.61	1200.86	**31.02**	33.55	37.43
kroC100	51.58	37.89	50.44	40.13	**37.85**	153.45	147.6	114.91	**86.21**	88.06	293.85	250.49	274.89	**179.66**	216.04
kroD100	132.6	130.6	121.79	**117.75**	146.72	63.35	103.79	**57.52**	68.15	99.68	44.7	48.29	**42.84**	58.33	47.48
kroE100	t.l.	3407.87	t.l.	**3376.34**	4545.15	44.97	**41.76**	46.38	42.0	70.86	**34.07**	34.12	39.01	42.03	52.24
rd100	**28.77**	29.23	35.35	32.04	36.53	**51.72**	66.83	56.69	56.82	64.63	84.59	**72.26**	109.85	76.34	91.18
eil101	21.66	12.74	26.45	**12.51**	12.67	48.5	31.43	28.28	22.05	**17.55**	9.0	11.09	**4.79**	8.35	10.27
lin105	t.l.	t.l.	t.l.	t.l.	t.l.	9499.53	7057.23	**423.4**	562.81	661.52	14074.3	13309.1	**1142.09**	1289.76	1321.56
pr107	t.l.	**86.48**	t.l.	88.59	104.62	t.l.	t.l.	t.l.	t.l.	t.l.	**6307.42**	6981.48	7468.42	6754.77	7863.77
pr124	t.l.	**113.99**	t.l.	115.2	255.0	t.l.	262.74	t.l.	**206.93**	211.31	329.6	**278.58**	332.22	313.39	725.42
bier127	68.37	**53.41**	66.93	67.82	96.37	**4654.12**	5327.09	10868.1	10620.2	13044.5	2932.07	3460.42	**1119.78**	1400.12	1627.7
ch130	217.93	**145.18**	213.49	160.96	186.64	**60.85**	63.05	74.91	68.12	94.7	**17.73**	19.19	26.27	33.46	31.25
pr136	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	13060.3	**4564.78**	4642.22	5149.9	t.l.	t.l.	5991.83	4290.11	**3913.4**
gr137	1120.93	**232.62**	1247.7	255.41	277.62	t.l.	t.l.	t.l.	t.l.	t.l.	804.57	814.21	**83.93**	105.26	88.84
pr144	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	**3369.23**	4506.69	4877.5	3987.22	4985.61
ch150	321.67	305.76	372.25	**272.98**	312.11	**132.36**	177.4	155.57	149.37	180.55	1180.97	1327.91	**198.65**	210.66	247.67
kroA150	512.45	259.44	612.28	261.09	**254.04**	**78.72**	101.3	83.51	99.13	98.09	t.l.	t.l.	t.l.	t.l.	t.l.
kroB150	379.28	315.61	369.33	**286.07**	531.03	1046.19	960.18	263.56	300.12	**256.38**	3259.84	5669.76	**359.08**	502.76	635.61
pr152	450.33	**362.71**	630.55	367.61	509.36	t.l.	t.l.	t.l.	t.l.	t.l.	2641.48	**1798.74**	3717.27	1904.74	2024.71
u159	t.l.	t.l.	t.l.	t.l.	t.l.	**328.3**	411.7	385.39	414.5	606.13	4153.0	**3066.63**	6066.08	3952.92	3176.01
rat195	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	800.68	885.05	**776.12**	1225.45	**1223.75**	1408.03	1488.77	1729.46
d198	10528.4	7601.72	11297.5	**7423.25**	8989.99	3777.76	**1974.04**	3746.0	2793.59	2962.89	t.l.	t.l.	4840.63	**4646.09**	4805.87
kroA200	1605.83	1446.99	**1295.99**	1425.59	1953.03	1885.84	1520.88	1021.45	997.23	**803.92**	t.l.	t.l.	t.l.	t.l.	t.l.
kroB200	1834.95	2026.48	**1554.23**	2002.76	2229.22	2582.52	2783.39	1560.88	**1398.61**	1400.64	7231.44	6869.73	**423.13**	532.67	429.05
gr202	1933.11	**1831.86**	2410.66	2637.85	2307.42	7021.03	5913.48	**5085.5**	5708.64	5159.16	t.l.	t.l.	t.l.	t.l.	t.l.
ts225	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
tsp225	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	4836.49	**4433.22**	4448.24
pr226	1958.39	1206.69	1929.79	1158.75	**1043.21**	t.l.	t.l.	t.l.	t.l.	t.l.	13495.0	t.l.	11637.0	**8540.59**	9861.3
gr229	**405.21**	809.32	413.72	935.03	949.68	5303.86	7355.38	4481.33	5993.76	**4004.14**	t.l.	t.l.	t.l.	t.l.	t.l.
gil262	885.14	830.1	**753.32**	834.5	848.15	t.l.	t.l.	446.67	**387.02**	390.47	t.l.	t.l.	11136.8	3927.83	**3848.21**
pr264	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	2767.16	1394.88	1518.26	1560.75	**1310.94**
a280	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	4951.5	**4324.35**	4371.21	t.l.	t.l.	t.l.	t.l.	t.l.
pr299	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	**5129.46**	5795.53	5800.57	t.l.	t.l.	t.l.	t.l.	t.l.
lin318	t.l.	t.l.	t.l.	t.l.	t.l.	11217.7	**6867.39**	10717.3	7372.96	11377.0	t.l.	t.l.	t.l.	**13249.4**	t.l.
rd400	12679.7	**11875.9**	t.l.	13019.4	t.l.	2960.33	3567.83	3901.39	3495.23	**2602.41**	t.l.	t.l.	t.l.	t.l.	t.l.
fl417	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
gr431	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
pr439	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	10052.4	8166.56	**7880.73**	t.l.	t.l.	t.l.	t.l.	t.l.
pcb442	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
d493	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
att532	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.

Table 5: Computation times (in seconds) on TSP generation 3 instances

5.4 Inequalities associated with the fastest sequences

Tables 6, 7, 8 and 9 give the number of inequalities of each type that were added using the fastest separation sequence for each instance. A dash means that the corresponding inequality was not part of the fastest

sequence. Without surprise, we observe that CY outnumbers by far all other inequalities with 75% of added inequalities on average. It is followed by LO (16%), 2M (5%), SC (4%), CC (3%), LC (2%), and CO (<1%). Although they are added in relatively small numbers, comb and cover inequalities remain essential most of the time, as it was shown in the previous section. However, it suggests that one should not allow too much time in the separation phase for those inequalities. That is, heuristic separation algorithms are appropriate since they can find enough violated inequalities to significantly reduce the expansion of the branching tree, with relatively small computation times.

α	0.25							0.50							0.75						
Instance	LO	CY	CC	LC	2M	SC	CO	LO	CY	CC	LC	2M	SC	CO	LO	CY	CC	LC	2M	SC	CO
st70	66	104	0	0	0	0	0	80	209	0	0	0	0	0	66	329	-	-	-	-	-
eil76	63	60	-	-	0	0	-	80	86	-	-	0	0	-	49	50	0	0	-	-	-
pr76	97	216	0	0	0	0	0	79	167	0	0	61	5	-	101	431	0	0	579	301	0
gr96	162	515	137	0	0	0	0	152	847	2	0	-	-	-	93	685	-	-	-	-	-
rat99	130	290	-	-	0	0	-	145	561	0	0	5	8	-	117	497	-	-	11	4	-
kroA100	190	635	0	0	0	0	-	152	655	0	0	3	1	-	122	556	0	0	60	0	-
kroB100	184	581	0	0	5	2	3	145	548	0	0	7	0	-	120	539	0	0	26	0	0
kroC100	190	685	283	0	2	4	-	193	899	-	-	-	-	-	136	604	-	-	-	-	-
kroD100	266	862	40	0	-	-	-	142	484	-	-	-	-	-	98	431	-	-	20	0	-
kroE100	146	437	4	0	-	-	-	144	473	-	-	23	3	-	103	396	-	-	24	1	-
rd100	138	472	-	-	-	-	-	133	528	-	-	-	-	-	103	514	-	-	-	-	-
eil101	136	305	-	-	0	0	-	121	329	-	-	-	-	-	57	197	-	-	7	0	-
lin105	167	491	-	-	-	-	-	185	1362	-	-	-	-	-	107	875	-	-	560	198	-
pr107	158	246	-	-	0	0	-	190	612	0	0	0	0	0	213	778	0	0	-	-	-
pr124	283	974	636	0	3	0	0	171	1055	-	-	-	-	-	149	1334	0	0	2	0	0
bier127	135	381	0	0	-	-	-	108	294	-	-	0	0	-	67	388	0	0	-	-	-
ch130	228	894	-	-	-	-	-	156	570	0	0	0	0	0	141	870	0	0	-	-	-
pr136	259	1124	-	-	-	-	-	342	2436	0	0	10	0	-	188	561	0	0	83	25	0
gr137	240	888	0	0	0	0	-	156	756	-	-	0	0	-	127	971	-	-	0	0	-
pr144	327	1176	-	-	0	0	-	302	1886	0	0	-	-	-	215	1455	-	-	-	-	-
ch150	256	852	0	0	-	-	-	196	732	0	0	-	-	-	177	790	-	-	438	106	-
kroA150	266	923	-	-	0	0	-	328	1754	-	-	2627	327	-	175	913	-	-	811	191	-
kroB150	277	1290	0	0	0	0	4	247	1388	0	0	0	0	2	163	776	0	0	2	0	-
pr152	317	1423	-	-	0	0	-	303	1635	-	-	-	-	-	253	1900	-	-	-	-	-
u159	279	2916	1064	0	-	-	-	292	2949	-	-	-	-	-	147	1864	-	-	-	-	-
rat195	306	1175	122	0	13	3	8	311	1233	0	0	-	-	-	266	1111	0	0	83	36	-
d198	321	1246	-	-	1	0	-	414	2674	-	-	302	325	-	196	1917	0	0	140	47	-
kroA200	399	2015	365	0	-	-	-	307	1621	0	0	40	61	-	167	1169	-	-	38	8	-
kroB200	342	1500	0	0	38	0	1	377	2074	0	0	-	-	-	199	1309	-	-	14	0	-
gr202	252	1173	-	-	-	-	-	226	1460	-	-	-	-	-	137	1139	-	-	183	22	-
ts225	704	12205	630	0	1331	685	485				t.l.							t.l.			
tsp225	429	2296	-	-	0	2	-	381	2445	0	0	87	31	-	271	2327	0	0	321	715	-
pr226	676	2615	1372	0	8	2	1	818	5141	-	-	-	-	-	440	4012	0	0	0	0	0
gr229	308	1691	0	0	7	15	0	275	2399	-	-	-	-	-	116	1560	0	0	25	20	-
gil262	550	2771	0	0	0	0	-	419	2416	0	0	-	-	-				t.l.			
pr264	337	341	0	0	0	0	0	509	1694	0	0	77	78	0	500	2039	-	-	0	0	-
a280	1025	13273	-	-	0	0	-	536	2550	-	-	0	0	-	392	2705	0	0	36	16	-
pr299	637	3503	0	0	-	-	-				t.l.				340	3909	-	-	61	2	-
lin318				t.l.				621	7337	0	0	401	49	7	315	3560	-	-	-	-	-
rd400	814	5329	0	0	-	-	-	590	5208	-	-	324	291	-	437	4049	-	-	2653	2287	-
gr431	260	1639	0	0	605	253	-	439	4669	-	-	425	950	-				t.l.			
pr439				t.l.				661	5013	-	-	38	33	-				t.l.			
d493	1619	13880	0	0	-	-	-	503	7205	-	-	507	0	-				t.l.			
att532	1021	8974	-	-	295	76	-	750	9368	0	0	767	600	0				t.l.			

Table 6: Number of inequalities of each type for TSP generation 1 instances

α	0.25							0.50							0.75						
Instance	LO	CY	CC	LC	2M	SC	CO	LO	CY	CC	LC	2M	SC	CO	LO	CY	CC	LC	2M	SC	CO
st70	69	108	-	-	0	0	-	91	232	0	1	0	0	0	69	254	0	10	174	91	-
eil76	82	153	-	-	-	-	-	122	380	0	11	4	14	-	115	339	0	13	-	-	-
pr76	122	412	174	26	0	0	-	146	678	-	-	-	-	-	135	734	-	-	450	882	-
gr96	219	872	-	-	-	-	-	176	1109	10	16	0	0	-	166	1136	-	-	36	14	-
rat99	139	379	-	-	-	-	-	222	996	0	57	-	-	-	152	594	-	-	-	-	-
kroA100	229	1013	288	31	0	0	14	228	998	-	-	-	-	-	222	1011	-	-	373	153	-
kroB100	146	462	0	11	-	-	-	196	880	-	-	-	-	-	146	869	-	-	47	8	-
kroC100	232	1008	400	67	48	27	23	261	1573	-	-	-	-	-	144	692	0	21	-	-	-
kroD100	279	1206	-	-	13	9	-	168	710	-	-	-	-	-	169	764	-	-	-	-	-
kroE100	162	360	-	-	-	-	-	184	715	-	-	0	0	-	189	880	-	-	-	-	-
rd100	182	953	-	-	13	0	-	204	810	-	-	6	9	-	165	675	0	65	1	10	-
eil101	123	309	-	-	0	0	-	141	417	-	-	-	-	-	125	512	-	-	308	535	-
lin105	102	224	-	-	-	-	-	209	1566	-	-	1401	1200	-	151	1046	0	171	-	-	-
pr107	242	791	0	27	-	-	-	231	1457	-	-	-	-	-	234	2331	0	221	36	97	-
pr124	285	1450	620	26	-	-	-	281	2665	0	71	-	-	-	223	2422	0	285	5	12	-
bier127	189	606	-	-	-	-	-	184	584	-	-	-	-	-	168	1103	-	-	589	332	-
ch130	285	1272	-	-	-	-	-	265	1175	0	27	-	-	-	145	951	-	-	-	-	-
pr136	195	793	-	-	-	-	-				t.l.				253	1678	0	146	-	-	-
gr137	229	1163	-	-	-	-	-	260	1525	-	-	10	10	-	258	2626	-	-	143	17	-
pr144	322	1546	690	0	-	-	-	324	2566	-	-	61	14	-	275	2197	0	303	-	-	-
ch150	260	976	-	-	-	-	-	371	2452	0	79	169	346	16	172	906	0	15	-	-	-
kroA150	385	1957	117	35	-	-	-	285	1295	-	-	-	-	-	357	3980	0	1142	2676	2408	-
kroB150	288	1307	0	5	0	6	-	449	2791	0	28	2	0	-	264	1873	0	44	-	-	-
pr152	387	2010	0	8	0	0	0	331	2169	0	140	-	-	-	3393	13518	-	-	-	-	-
u159	324	2334	863	58	1	19	-	383	4454	0	95	36	4	34	345	5925	0	371	-	-	-
rat195	336	1865	73	5	11	0	2	280	1573	0	2	17	2	-	237	1531	0	1	251	753	-
d198	319	1616	-	-	-	-	-	478	3780	-	-	202	354	-				t.l.			
kroA200	553	3270	275	65	42	196	2	525	3020	0	146	623	1842	40				t.l.			
kroB200	413	2115	211	5	19	7	-	507	2910	-	-	-	-	-				t.l.			
gr202	315	1847	0	23	3	6	-	319	2045	0	81	656	570	4	273	2158	-	-	-	-	-
ts225	475	2857	-	-	-	-	-	440	3471	0	17	-	-	-	494	4044	0	69	1556	810	-
tsp225	511	2587	914	34	-	-	-	457	4108	0	0	-	-	-	344	2696	0	209	-	-	-
pr226	693	3501	0	5	0	0	2	862	7520	139	706	106	207	103	512	7592	0	109	14	10	7
gr229				t.l.				429	4270	0	45	819	1416	-	328	3385	-	-	-	-	-
gil262				t.l.				417	2546	0	8	0	0	-	423	3517	0	207	2917	1114	9
pr264	644	3625	0	128	88	84	16				t.l.				529	3778	0	149	-	-	-
a280	878	5842	0	85	-	-	-				t.l.				371	3476	0	11	-	-	-
lin318				t.l.				616	6909	0	13	-	-	-	554	6716	0	69	53	163	-
rd400				t.l.				717	7066	-	-	210	242	-	493	6331	0	6	1014	1284	-
fl417	2409	11346	-	-	0	0	-				t.l.							t.l.			
gr431	315	1920	-	-	876	491	-				t.l.							t.l.			
pr439	1157	8820	-	-	28	17	-				t.l.							t.l.			
pcb442	1198	8276	-	-	5	0	-				t.l.							t.l.			
att532	1267	8519	0	60	31	5	15				t.l.							t.l.			

Table 7: Number of inequalities of each type for TSP generation 2 instances

α	0.25							0.50							0.75						
Instance	LO	CY	CC	LC	2M	SC	CO	LO	CY	CC	LC	2M	SC	CO	LO	CY	CC	LC	2M	SC	CO
st70	154	685	73	10	0	16	11	139	522	-	-	-	-	-	128	441	0	13	273	160	-
eil76	210	692	41	17	-	-	-	180	519	-	-	1	3	-	50	111	-	-	-	-	-
pr76	190	1493	87	24	5	3	-	144	721	-	-	2	0	-	137	863	-	-	868	79	-
gr96	299	1350	344	58	-	-	-	223	1389	0	24	27	6	-	192	1886	-	-	-	-	-
rat99	349	2447	319	6	47	13	151	270	1109	0	33	-	-	-	171	729	0	5	32	15	3
kroA100	296	1407	50	19	8	0	31	245	960	-	-	-	-	-	219	1149	-	-	94	103	-
kroB100	436	2072	53	45	0	6	49	236	873	-	-	20	13	-	152	743	-	-	124	70	-
kroC100	246	1127	446	18	0	0	6	248	1344	0	22	8	0	-	207	1164	0	61	48	51	-
kroD100	329	1403	111	67	46	2	-	232	1024	-	-	34	50	-	191	704	-	-	14	35	-
kroE100	551	4141	341	19	4	25	-	310	1278	0	12	-	-	-	207	876	-	-	-	-	-
rd100	273	911	-	-	-	-	-	204	953	-	-	-	-	-	182	977	0	98	-	-	-
eil101	199	535	12	2	0	6	-	147	409	0	13	27	21	2	68	230	-	-	4	0	-
lin105	t.l.							216	1374	-	-	772	525	-	216	1403	-	-	1793	1275	-
pr107	349	1177	1475	46	-	-	-	t.l.							304	1640	-	-	-	-	-
pr124	306	1540	675	5	-	-	-	599	4279	119	46	34	37	-	261	2462	0	7	-	-	-
bier127	268	930	8	23	-	-	-	326	3073	-	-	-	-	-	196	1144	-	-	614	1128	-
ch130	356	1748	80	6	-	-	-	231	1078	-	-	-	-	-	118	501	-	-	-	-	-
pr136	t.l.							442	2855	-	-	70	421	-	334	1867	0	404	219	1042	23
gr137	326	1805	741	99	-	-	-	t.l.							211	953	-	-	35	13	-
pr144	t.l.							t.l.							280	2517	-	-	-	-	-
ch150	485	2815	51	24	0	0	-	361	1996	-	-	-	-	-	357	1473	-	-	89	84	-
kroA150	525	2837	30	48	5	0	10	302	1392	-	-	-	-	-	t.l.						
kroB150	418	2327	33	82	133	8	-	391	2611	0	13	123	17	3	269	1422	-	-	236	77	-
pr152	451	2449	1254	32	-	-	-	t.l.							336	3087	0	107	-	-	-
u159	t.l.							523	4900	-	-	-	-	-	308	3934	0	101	-	-	-
rat195	t.l.							667	3651	0	26	90	92	8	503	2471	0	80	-	-	-
d198	556	3563	1236	665	214	531	-	432	2626	0	473	-	-	-	371	3066	0	298	339	1765	-
kroA200	740	4899	-	-	4	15	-	444	2689	0	30	223	451	6	t.l.						
kroB200	886	5335	-	-	0	0	-	598	3321	0	46	162	235	-	350	1682	-	-	24	21	-
gr202	441	2784	224	295	-	-	-	381	2791	-	-	1060	1869	-	t.l.						
tsp225	t.l.							t.l.							491	2379	0	275	732	830	-
pr226	802	4047	1503	5	42	4	4	t.l.							691	9441	0	72	19	41	-
gr229	571	4271	-	-	-	-	-	543	5724	0	130	131	33	41	t.l.						
gil262	1074	6470	-	-	8	2	-	551	3713	0	0	92	23	-	483	3564	0	315	393	301	36
pr264	t.l.							t.l.							1195	5272	0	11	2	1	1
a280	t.l.							1204	6829	0	89	512	295	-	t.l.						
pr299	t.l.							1706	8734	-	-	70	39	-	t.l.						
lin318	t.l.							661	7157	0	160	-	-	-	585	6957	0	161	305	794	-
rd400	1868	15436	0	22	-	-	-	846	6581	0	21	151	130	21	t.l.						
pr439	t.l.							1399	10183	0	24	279	382	2	t.l.						

Table 8: Number of inequalities of each type for TSP generation 3 instances

α	0.25							0.50							0.75						
Instance	LO	CY	CC	LC	2M	SC	CO	LO	CY	CC	LC	2M	SC	CO	LO	CY	CC	LC	2M	SC	CO
att48	59	115	33	0	0	0	-	48	74	0	0	-	-	-	29	67	0	0	0	0	-
eil51	44	30	-	-	-	-	-	71	182	-	-	-	-	-	84	271	0	15	-	-	-
eilA76	108	246	22	6	-	-	-	72	236	0	0	0	4	5	80	172	0	9	21	8	-
eilA101	146	441	71	3	0	0	1	176	558	0	4	0	2	-	75	190	0	6	-	-	-
gil262	910	3458	0	32	-	-	-	549	3670	0	17	-	-	-	286	2558	0	26	69	50	-

Table 9: Number of inequalities of each type for VRP instances

5.5 General results associated with the fastest separation sequences

General statistics for each instance and each α value, using the fastest separation sequence, are given in Tables 10, 11, 12 and 13. The columns correspond to:

- TIME: the total computation time;
- NDS: the number of branching nodes explored;
- CUTS: the number of cuts added to the LP;
- GAP: the gap (%) between the upper and lower bounds at the root;
- %UB: the quality of the upper bound obtained at the root;
- OPT: the optimal value.

For instances still unsolved after 4 hours (t.l.), we give the best feasible solution in the OPT column and the gap (%) between the best lower bound and the best upper bound (value of the best known feasible solution) in the GAP column.

The results confirm that TSP instances of generation 3 are the most difficult to solve while generation 1 instances are the easiest. The average computation time increases from 757 to 1563 seconds between generation 1 and 3. Moreover, an average of 91 branching nodes are explored for generation 1 instances against 855 nodes for generation 3 instances. The quality of the upper and lower bounds at the root is also worse for generation 3 (average gap of 10.4%) as compared to generations 1 (7.4%) and 2 (6.2%). We observe that the average computation time is about the same for all three values of α even if the average number of explored nodes rise from 237 to 452 between $\alpha = 0.25$ and $\alpha = 0.75$. This increase might be due to the heuristics used to find feasible solutions (see sections 4.2 and 4.5) that seem to find better solutions when α is high. Actually, the average gap at the root is 10.9%, 5.7% and 4.5% for $\alpha = 0.25$, 0.50 and 0.75, respectively. Moreover, the quality of the upper bound at the root node is 6.2%, 4.0% and 3.1% for $\alpha = 0.25$, 0.50 and 0.75, respectively. Among the 455 instances considered, 30 where solved at the root node of the branching tree and 85 where still unsolved after 4 hours of computation. For those instances, the final gap between the lower bound and the best feasible solution found is of 4.3%, on average. One should also note that 10 unsolved problems show a final gap under 0.1%.

α	0.25						0.50						0.75					
Instance	TIME	NDS	CUTS	GAP	%UB	OPT	TIME	NDS	CUTS	GAP	%UB	OPT	TIME	NDS	CUTS	GAP	%UB	OPT
att48	0.52	3	207	10.15	9.84	6896	0.45	3	122	3.87	3.65	11734	0.44	1	96	0.0	0.0	20685
eil51	0.29	1	74	0.0	0.0	76	1.3	37	253	2.48	0.0	161	3.03	69	370	2.33	1.55	254
eilA76	4.11	24	382	14.15	11.32	94	2.97	13	317	4.5	4.0	192	4.88	32	290	4.75	4.75	321
eilA101	7.33	12	662	18.58	17.7	93	8.53	5	740	0.97	0.49	205	4.99	21	271	0.59	0.3	336
gil262	425.49	35	4400	1.03	0.26	387	534.61	31	4236	2.79	2.43	803	278.45	26	2989	0.84	0.76	1305

Table 10: Statistics on the resolution for VRP instances

α	0.25						0.50						0.75					
Instance	TIME	NDS	CUTS	GAP	%UB	OPT	TIME	NDS	CUTS	GAP	%UB	OPT	TIME	NDS	CUTS	GAP	%UB	OPT
st70	0.6	1	170	0.0	0.0	129	0.85	1	289	0.0	0.0	260	3.14	22	395	10.71	10.08	428
eil76	0.8	1	123	0.0	0.0	102	0.94	1	166	0.0	0.0	235	0.78	1	99	0.0	0.0	335
pr76	1.31	1	313	0.0	0.0	23450	2.39	5	312	0.31	0.02	41248	109.56	389	1412	5.22	3.89	64142
gr96	5.21	1	814	0.0	0.0	10460	38.05	125	1001	13.54	12.51	20688	12.5	11	778	3.33	3.03	31343
rat99	2.87	1	420	0.0	0.0	293	14.3	29	719	2.2	1.53	581	11.67	9	629	1.95	1.83	856
kroA100	13.07	25	825	18.73	14.42	4931	9.11	7	811	5.6	5.32	9184	10.83	9	738	3.7	3.39	14379
kroB100	10.16	13	775	10.23	9.18	4305	5.72	1	700	0.0	0.0	9096	7.42	1	685	0.0	0.0	14680
kroC100	16.99	29	1164	8.78	6.27	4964	18.26	19	1092	8.11	6.62	9457	10.11	7	740	3.65	3.29	13892
kroD100	16.11	23	1168	15.98	10.46	4762	6.3	9	626	0.64	0.4	8719	6.99	1	549	0.0	0.0	14058
kroE100	6.75	13	587	1.28	0.0	3905	6.87	5	643	1.27	1.18	9097	7.12	3	524	0.57	0.5	14622
rd100	4.57	3	610	2.97	2.97	1438	6.53	7	661	4.56	3.8	3168	9.52	19	617	3.96	3.39	5094
eil101	4.59	8	441	1.77	0.88	112	4.36	5	450	5.33	4.92	232	3.1	2	261	1.26	1.26	391
lin105	7.79	5	658	25.12	8.61	2718	168.02	735	1547	11.38	8.09	5920	103.17	93	1740	2.18	1.61	9120
pr107	2.34	1	404	0.0	0.0	8678	6.87	1	802	0.0	0.0	18311	16.23	9	991	2.1	1.96	36292
pr124	30.38	15	1896	6.37	1.99	14325	13.3	1	1226	0.0	0.0	22998	42.17	7	1485	6.15	5.79	39008
bier127	6.5	7	516	1.34	0.46	10986	4.49	1	402	0.0	0.0	26347	11.1	19	455	3.14	2.8	49680
ch130	20.4	31	1122	9.96	7.94	1183	8.64	1	726	0.0	0.0	2408	69.65	245	1011	4.33	3.61	3876
pr136	20.21	3	1383	13.63	13.53	21116	71.88	13	2788	27.28	13.24	46167	24.83	1	857	0.0	0.0	67910
gr137	17.58	9	1128	9.97	9.79	17723	10.62	1	912	0.0	0.0	29575	23.3	1	1098	0.0	0.0	44100
pr144	29.09	1	1503	0.0	0.0	14327	84.26	23	2188	14.68	13.43	27424	138.9	93	1670	13.03	11.55	41136
ch150	20.31	11	1108	3.1	1.7	1332	22.95	27	928	0.61	0.29	2760	132.18	57	1511	1.52	1.09	4366
kroA150	21.35	11	1189	4.03	3.31	5786	2137.76	2003	5036	6.36	4.49	11496	329.77	549	2090	0.65	0.12	17395
kroB150	39.98	9	1571	4.29	3.53	5896	36.53	5	1637	1.11	1.09	11357	25.68	5	941	1.45	1.42	16879
pr152	44.14	5	1740	0.55	0.0	20029	68.85	33	1938	5.71	5.29	36333	161.66	197	2153	1.19	1.0	51826
u159	90.94	23	4259	0.85	0.63	9266	570.01	251	3241	8.37	7.2	18511	48.3	1	2011	0.0	0.0	27096
rat195	102.23	25	1627	2.62	2.45	558	156.26	56	1544	5.62	5.28	1112	211.16	19	1496	2.81	2.63	1665
d198	46.58	1	1568	0.0	0.0	4994	1366.88	359	3715	6.26	5.69	6913	222.57	19	2300	19.61	11.93	9091
kroA200	197.99	27	2779	2.71	2.32	6154	118.79	26	2029	4.85	4.49	12372	63.84	3	1382	0.49	0.46	19211
kroB200	64.5	9	1881	2.37	1.76	6016	351.33	109	2451	9.7	8.67	12338	61.29	1	1522	0.0	0.0	18898
gr202	30.33	1	1425	0.0	0.0	8221	328.51	311	1686	4.51	3.94	13790	108.26	13	1481	4.18	4.08	20914
ts225	6531.68	1933	16040	13.56	11.68	29121	t.l.			2.55		57995	t.l.			0.46		85949
tsp225	187.05	19	2727	14.42	12.91	924	317.29	79	2944	26.66	11.43	1721	1522.61	388	3634	4.76	4.54	2629
pr226	410.39	29	4674	65.22	7.78	20158	5429.52	351	5959	68.25	15.8	36720	573.84	9	4452	59.24	27.46	47142
gr229	83.23	13	2021	5.98	5.72	18823	154.54	9	2674	3.05	2.95	39875	143.54	7	1721	2.45	2.36	64959
gil262	179.51	7	3321	3.58	3.22	541	165.14	18	2835	4.28	4.09	986	t.l.			0.39		1552
pr264	24.12	1	678	0.0	0.0	9232	532.23	13	2358	4.71	4.63	22644	177.51	3	2539	43.25	13.38	34346
a280	1765.66	271	14298	21.12	10.97	649	303.65	17	3086	4.21	4.13	1231	461.14	7	3149	10.7	8.82	1841
pr299	562.18	17	4140	14.08	13.72	11209	t.l.			2.89		23089	672.98	3	4312	5.47	5.45	32674
lin318	t.l.			2.68		8787	2355.21	29	8415	12.9	12.63	15913	1860.79	285	3875	13.86	13.47	25288
rd400	483.4	1	6143	0.0	0.0	3084	2110.73	45	6413	4.67	4.48	6284	11466.7	217	9426	2.81	2.59	9906
fl417	t.l.			30.68		1975	t.l.			11.64		5754	t.l.			3.88		8149
gr431	883.06	51	2757	2.74	2.46	14641	14285.7	583	6483	2.96	2.42	35222	t.l.			1.98		74952
pr439	t.l.			7.59		21334	1483.28	8	5745	1.11	1.09	35297	t.l.			5.21		60803
pcb442	t.l.			0.63		11067	t.l.			0.01		22281	t.l.			0.31		34305
d493	11933.1	320	15499	37.02	19.33	9355	1943.26	44	8215	1.19	1.16	13582	t.l.			6.22		22338
att532	4521.65	14	10366	4.3	4.17	3812	10280.1	69	11485	3.68	3.54	8943	t.l.			7.60		17406

Table 11: Statistics on the resolution for TSP generation 1 instances

α	0.25						0.50						0.75					
Instance	TIME	NDS	CUTS	GAP	%UB	OPT	TIME	NDS	CUTS	GAP	%UB	OPT	TIME	NDS	CUTS	GAP	%UB	OPT
st70	0.96	5	177	4.65	3.88	124	1.31	3	324	1.98	1.98	247	7.5	19	598	5.63	4.93	405
eil76	1.38	1	235	0.0	0.0	99	6.44	31	531	6.64	5.21	200	5.21	31	467	1.59	0.95	312
pr76	5.87	29	734	7.03	5.77	22157	22.13	173	824	5.76	5.41	38330	188.18	977	2201	1.06	0.0	58419
gr96	34.76	105	1091	10.15	4.79	10991	34.6	95	1311	10.39	9.07	19380	62.01	117	1352	1.97	1.38	28725
rat99	17.75	159	518	12.12	9.76	268	61.5	196	1275	9.43	7.83	518	24.67	129	746	1.3	0.65	767
kroA100	38.38	55	1575	18.87	13.44	4716	29.7	79	1226	7.64	4.24	8519	96.42	141	1759	4.34	3.87	13115
kroB100	7.01	11	619	5.31	2.31	3590	41.7	173	1076	8.92	7.0	7794	35.41	71	1070	2.56	2.18	13441
kroC100	68.63	103	1805	3.47	0.97	4881	41.16	35	1834	9.52	6.96	9060	29.02	77	857	2.17	1.48	12732
kroD100	75.45	139	1507	14.77	10.11	4777	30.74	113	878	12.2	11.36	8267	40.37	147	933	3.37	3.2	12897
kroE100	12.23	79	522	14.07	12.76	3397	17.9	33	899	14.05	13.73	7644	91.46	249	1069	2.43	1.98	12803
rd100	19.87	25	1148	7.16	6.97	1508	22.29	73	1029	3.08	2.23	2892	44.44	115	916	0.64	0.37	4350
eil101	3.73	1	432	0.0	0.0	108	9.11	23	558	3.69	2.76	211	71.52	89	1480	2.55	1.7	347
lin105	5.48	35	326	2.72	2.18	2515	716.02	1397	4376	6.97	5.09	5614	143.45	391	1368	8.71	7.14	8604
pr107	22.7	31	1060	0.59	0.11	8322	76.69	321	1688	9.4	9.23	26372	212.35	305	2919	3.19	2.85	33592
pr124	130.58	77	2381	14.45	10.72	14261	162.39	93	3017	1.01	0.51	23150	390.93	147	2947	8.74	8.26	37407
bier127	50.33	173	795	6.84	3.79	10525	37.55	113	768	1.27	0.95	24478	298.98	401	2192	4.63	3.82	48015
ch130	79.37	139	1557	9.93	5.7	1159	83.66	147	1467	7.16	6.45	2220	37.41	43	1096	2.07	1.91	3552
pr136	19.61	39	988	3.85	2.68	18143	t.l.			1.64		40241	308.91	371	2077	5.79	5.63	58571
gr137	75.21	131	1392	8.96	7.53	17252	366.15	281	1805	2.19	1.72	28242	441.08	309	3044	1.73	1.4	42644
pr144	56.94	5	2558	1.75	1.71	13991	284.38	155	2965	3.96	3.51	27073	963.33	705	2775	6.88	5.53	40589
ch150	35.57	31	1236	10.62	9.96	1238	541.81	285	3433	5.1	3.66	2476	36.02	31	1093	8.03	7.7	3922
kroA150	172.63	81	2494	16.71	14.74	5306	60.85	43	1580	9.26	8.85	9968	6748.62	3793	10563	1.81	0.84	15658
kroB150	56.77	37	1606	0.2	0.0	4958	469.95	115	3270	7.91	6.27	10278	238.58	123	2181	3.33	2.92	14971
pr152	121.46	47	2405	1.17	0.96	19864	249.75	173	2640	2.61	2.28	34474	4299.57	5887	16911	6.44	5.17	50520
u159	320.51	147	3599	13.07	9.96	8953	763.28	169	5006	4.6	3.22	17161	2569.05	535	6641	1.92	1.75	25377
rat195	129.58	31	2292	2.89	2.7	505	112.81	13	1874	3.24	3.14	988	543.85	25	2773	5.88	5.63	1510
d198	141.14	109	1935	7.84	7.65	4819	2579.62	470	4814	2.44	2.22	6653	t.l.			0.01		8835
kroA200	1469.1	109	4403	15.91	14.07	5898	2278.69	386	6196	6.98	5.91	11219	t.l.			0.53		17305
kroB200	223.31	55	2770	16.4	14.25	5490	415.38	175	3417	12.98	12.53	11250	t.l.			0.05		17210
gr202	169.37	95	2194	3.31	2.81	7930	753.52	373	3675	3.05	2.64	12804	636.21	450	2431	2.06	1.79	19165
ts225	576.52	207	3332	9.43	7.8	27478	907.77	127	3928	19.69	18.86	53102	2078.69	301	6973	4.86	4.33	79841
tsp225	341.86	73	4046	8.86	8.42	848	1803.93	351	4565	7.09	6.32	1585	670.7	315	3249	2.54	2.46	2416
pr226	498.88	23	4201	3.07	1.68	19956	8186.06	652	9643	13.96	11.15	36190	1846.79	137	8244	1.4	1.34	46096
gr229	t.l.			1.45		17215	3478.96	565	6979	4.58	3.97	35856	1254.23	319	3713	0.73	0.71	57295
gil262	t.l.			1.20		500	165.21	21	2971	0.12	0.12	865	4055.05	717	8187	2.24	1.95	1356
pr264	3447.13	213	4585	3.73	2.97	8721	t.l.			5.01		23660	1311.34	269	4456	0.3	0.24	33247
a280	1413.6	513	6805	0.7	0.52	572	t.l.			1.92		1143	561.59	20	3858	6.86	6.8	1658
pr299	t.l.			0.06		10292	t.l.			3.37		20613	t.l.			0.27		29980
lin318	t.l.			3.52		8275	3394.98	43	7538	1.82	1.49	14909	4222.08	390	7555	6.78	6.77	23571
rd400	t.l.			0.04		2784	3102.53	83	8235	3.23	2.78	5590	6745.16	107	9128	1.65	1.49	8915
fl417	5488.45	37	13755	7.56	6.92	1748	t.l.			13.95		5971	t.l.			0.01		7647
gr431	1882.38	241	3602	2.29	1.94	13555	t.l.			0.03		31725	t.l.			1.94		69195
pr439	10242.4	261	10022	0.33	0.24	18214	t.l.			0.15		33110	t.l.			8.89		61024
pcb442	4992.98	71	9479	0.7	0.57	9548	t.l.			0.02		19165	t.l.			2.94		31462
d493	t.l.			1.16		8832	t.l.			1.21		12835	t.l.			2.24		20014
att532	9026.86	253	9897	2.49	2.16	3536	t.l.			1.42		8213	t.l.			4.07		15535

Table 12: Statistics on the resolution for TSP generation 2 instances

α	0.25						0.50						0.75					
Instance	TIME	NDS	CUTS	GAP	%UB	OPT	TIME	NDS	CUTS	GAP	%UB	OPT	TIME	NDS	CUTS	GAP	%UB	OPT
st70	13.05	47	949	15.67	8.76	198	7.71	24	661	6.19	4.64	308	20.62	41	1015	1.82	1.14	434
eil76	32.68	175	960	19.86	10.64	126	9.05	24	703	4.27	3.32	204	1.71	7	161	2.8	2.8	312
pr76	1121.31	7385	1802	21.3	9.72	29811	24.02	159	867	1.11	0.25	42200	147.24	711	1947	0.63	0.0	63348
gr96	137.2	347	2051	13.31	9.5	15384	50.48	99	1669	8.62	8.14	22491	272.77	717	2078	4.29	3.5	31844
rat99	922.85	1281	3332	25.21	15.04	401	55.05	101	1412	8.17	5.39	579	37.84	45	955	0.37	0.12	808
kroA100	254.08	623	1811	22.07	9.44	5803	17.47	21	1205	1.97	1.4	8325	117.14	255	1565	8.94	7.7	13241
kroB100	200.34	265	2661	30.59	11.07	6434	42.17	101	1142	4.31	3.68	8768	31.02	53	1089	0.19	0.08	13447
kroC100	37.85	51	1843	16.84	12.52	5714	86.21	169	1622	6.44	4.61	9283	179.66	563	1531	1.64	1.03	13156
kroD100	117.75	249	1958	18.92	14.44	5936	57.52	167	1340	7.59	5.56	8998	42.84	125	944	0.24	0.0	12731
kroE100	3376.34	4743	5081	28.54	14.66	6968	41.76	37	1600	12.84	8.03	9313	34.07	113	1083	0.63	0.31	12740
rd100	28.77	59	1184	5.09	4.26	2068	51.72	157	1157	4.69	4.06	3377	72.26	165	1257	3.27	2.94	4925
eil101	12.51	19	754	13.64	12.12	116	17.55	41	619	1.33	0.89	223	4.79	11	302	0.53	0.53	373
lin105	t.l.			7.37		4952	423.4	676	2887	2.47	0.88	6547	1142.09	1879	4687	3.28	2.32	9482
pr107	86.48	159	3047	8.89	4.41	15553	t.l.			0.25		27198	6307.42	13873	1944	7.77	6.69	34064
pr124	113.99	79	2526	25.77	10.77	17479	206.93	129	5114	10.22	6.6	26375	278.58	113	2730	5.49	3.85	37799
bier127	53.41	115	1229	11.7	10.99	20912	4654.12	13859	3399	8.41	6.86	42358	1119.78	1787	3082	1.7	1.3	70777
ch130	145.18	111	2190	29.76	22.02	1381	60.85	109	1309	5.0	3.88	2305	17.73	17	619	1.38	1.3	3719
pr136	t.l.			1.56		40241	4564.78	3097	3788	3.72	3.0	42179	3913.4	2535	3889	1.28	0.94	60109
gr137	232.62	249	2971	13.45	12.5	21810	t.l.			2.18		34023	83.93	51	1212	0.86	0.78	44203
pr144	t.l.			4.44		22462	t.l.			4.98		30033	3369.23	2139	2797	10.87	9.71	38267
ch150	272.98	141	3375	27.65	16.63	1725	132.36	32	2357	1.55	1.29	2675	198.65	159	2003	0.48	0.28	3980
kroA150	254.04	103	3455	15.16	7.98	6102	78.72	69	1694	0.3	0.11	9409	t.l.			0.42		15801
kroB150	286.07	179	3001	10.89	8.15	6276	256.38	95	3158	8.47	7.89	10392	359.08	389	2004	2.57	2.2	15520
pr152	362.71	203	4186	9.61	8.88	26410	t.l.			4.61		40937	1798.74	937	3530	14.9	14.21	51130
u159	t.l.			0.20		12619	328.3	95	5423	3.25	2.8	17631	3066.63	1091	4343	0.71	0.46	26524
rat195	t.l.			13.02		722	776.12	177	4534	3.69	3.01	999	1223.75	613	3054	6.84	6.65	1471
d198	7423.25	3136	6765	8.66	8.31	5474	1974.04	757	3531	2.24	2.11	7388	4646.09	1495	5839	7.35	7.01	10241
kroA200	1295.99	181	5658	11.84	9.08	6975	803.92	229	3843	0.75	0.5	11987	t.l.			2.00		18659
kroB200	1554.23	167	6221	20.33	10.89	6988	1398.61	230	4362	1.73	1.51	10752	423.13	307	2077	1.4	1.23	16920
gr202	1831.86	1301	3744	7.67	7.22	8832	5085.5	2477	6101	3.91	3.68	14377	t.l.			0.09		21506
ts225	t.l.			23.57		40971	t.l.			0.69		53414	t.l.			1.45		80656
tsp225	t.l.			12.35		1198	t.l.			0.12		1649	4433.22	1151	4707	6.01	5.82	2474
pr226	1043.21	69	6407	5.11	4.62	27424	t.l.			0.77		39091	8540.59	829	10264	4.29	4.25	45451
gr229	405.21	117	4842	15.33	15.11	25258	4004.14	405	6602	2.06	1.89	46791	t.l.			0.46		76314
gil262	753.32	30	7554	9.77	9.31	594	387.02	12	4379	0.62	0.41	961	3848.21	675	5092	1.99	1.39	1486
pr264	t.l.			25.21		17932	t.l.			4.72		23264	1310.94	87	6482	0.88	0.83	29928
a280	t.l.			14.38		786	4324.35	185	8929	1.72	1.63	1084	t.l.			0.94		1596
pr299	t.l.			19.97		16278	5129.46	189	10549	1.14	1.06	20317	t.l.			1.24		30269
lin318	t.l.			5.57		10816	6867.39	831	7978	1.42	1.21	16401	13249.4	683	8802	4.12	4.05	25892
rd400	11875.9	262	17326	11.56	9.33	3050	2602.41	111	7750	0.49	0.38	5700	t.l.			0.58		9181
fl417	t.l.			11.92		4295	t.l.			8.59		5740	t.l.			0.99		6557
gr431	t.l.			19.93		36760	t.l.			2.27		56484	t.l.			2.94		93835
pr439	t.l.			0.75		22222	7880.73	87	12269	6.85	6.54	35771	t.l.			6.58		63974
pcb442	t.l.			12.51		12836	t.l.			0.46		19632	t.l.			1.41		30631
d493	t.l.			0.13		9360	t.l.			0.02		13480	t.l.			3.14		21469
att532	t.l.			24.61		7647	t.l.			0.94		10258	t.l.			4.13		17380

Table 13: Statistics on the resolution for TSP generation 3 instances

Conclusion

We have adapted inequalities from the knapsack polytope and the TS polytope to implement the first branch-and-cut procedure for the PCTSP. We have also presented separation algorithms for those inequalities as well as a heuristic procedure to find feasible solutions to the PCTSP. Our branch-and-cut algorithm has been tested on problems generated from TSP and VRP instances of the TSPLIB [26] ranging from 70 to 532 vertices. Our results show that subtour elimination and logical inequalities alone are not sufficient to tackle hard problems. Whereas adding cover inequalities appear to be enough for problems with small minimum collected prize $\bar{p}$ ($\alpha = 0.25$), the comb inequalities should be added when $\bar{p}$ is large ($\alpha = 0.75$). For intermediate values of $\bar{p}$ ($\alpha = 0.50$), both the cover and comb inequalities are required to obtain good results.

Acknowledgements

Financial support for this work was provided by the Canadian Natural Sciences and Engineering Research Council (NSERC) and by the Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT). This support is gratefully acknowledged.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[2] B. Awerbuch, Y. Azar, A. Blum, and S. Vempala. New approximation guarantees for minimum-weight k-trees and prize-collecting salesman. *SIAM Journal on Computing*, 28:254–262, 1998.

[3] E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 8:146–164, 1975.

[4] E. Balas. The prize collecting traveling salesman problem. *Networks*, 19:621–636, 1989.

[5] E. Balas. The prize collecting traveling salesman problem: II. Polyhedral results. *Networks*, 25:199–216, 1995.

[6] E. Balas. The prize collecting traveling salesman problem and its applications. In G. Gutin and A. P. Punnen, editors, *The Traveling Salesman Problem and its Variations*, pages 663–695. Kluwer Academic Publishers, 2002.

[7] E. Balas and E. Zemel. Facets of the knapsack polytope from minimal covers. *SIAM Journal of Applied Mathematics*, 34:119–148, 1978.

[8] V. Chvátal. Edmonds polytopes and weakly hamiltonian graphs. *Mathematical Programming*, 5:29–40, 1973.

[9] M. Dell'Amico, F. Maffioli, and A. Sciomachen. A lagrangian heuristic for the prize collecting travelling salesman problem. *Annals of Operations Research*, 81:289–305, 1998.

[10] M. Dell'Amico, F. Maffioli, and P. Värbrand. On prize-collecting tours and the asymmetric travelling salesman problem. *International Transactions in Operational Research*, 2:297–308, 1995.

[11] D. Feillet, P. Dejax, and M. Gendreau. Traveling salesman problems with profits. *Transportation Science*, 39:188–205, 2005.

[12] M. Fischetti, J. J. Salazar-González, and P. Toth. Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, 10:133–148, 1998.

[13] M. Fischetti, J. J. Salazar-González, and P. Toth. The generalized traveling salesman and orienteering problems. In G. Gutin and A. P. Punnen, editors, *The Traveling Salesman Problem and its Variations*, pages 609–662. Kluwer Academic Publishers, 2002.

[14] M. Fischetti and P. Toth. An additive approach for the optimal solution of the prize-collecting travelling salesman problem. In B. L. Golden and A. A. Assad, editors, *Vehicule Routing: Methods and Studies*, pages 319–343. Elsevier Science Publishers, B.V. North-Holland, 1988.

[15] M. Gendreau, A. Hertz, and G. Laporte. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40:1086–1094, 1992.

[16] M. Gendreau, G. Laporte, and F. Semet. A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks*, 32:263–273, 1998.

[17] M. Grötschel and M. W. Padberg. On the symmetric travelling salesman problem I: inequalities. *Mathematical Programming*, 16:265–280, 1979.

[18] M. Grötschel and M. W. Padberg. On the symmetric travelling salesman problem II: lifting theorems and facets. *Mathematical Programming*, 16:281–302, 1979.

[19] M. Jünger, G. Reinelt, and G. Rinaldi. The traveling salesman problem. In M. O. Balls, T. L. Magnanti, C. L. Monma, and G. Nemhauser, editors, *Handbooks in Operations Research and Management Science, Network Models*, volume 7, pages 225–330. North Holland, Amsterdam, 1995.

[20] S. Kataoka and S. Morito. An algorithm for single constraint maximum collection problem. *Journal of the Operations Research Society of Japan*, 31:515–530, 1988.

[21] G. Laporte and S. Martello. The selective travelling salesman problem. *Discrete Applied Mathematics*, 26:193–207, 1990.

[22] A. C. Leifer and M. B. Rosenwein. Strong linear programming relaxations for the orienteering problem. *European Journal of Operational Research*, 73:517–523, 1994.

[23] S. Martello and P. Toth. An upper bound for the zero-one knapsack problem and a branch and bound algorithm. *European Journal of Operational Research*, 1:169–175, 1977.

[24] S. Martello and P. Toth. *Knapsack Problems, Algorithms and Computer Implementations*. John Wiley and Sons, 1990.

[25] M. W. Padberg and G. Rinaldi. Facet identification for the symmetric traveling salesman polytope. *Mathematical Programming*, 47:219–257, 1990.

[26] G. Reinelt. TSPLIB, A traveling salesman problem library. *ORSA Journal on Computing*, 3:376–384, 1991.

[27] T. Tsiligirides. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35:797–809, 1984.